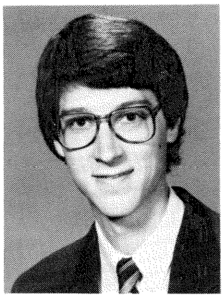
SIMULATION 44:5, 219-224.

ISSN 0037-5497/85 $2.00+.10
Printed in the United States of America

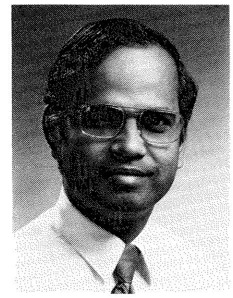
Key words: graphical analysis, queuing networks, simulation

# A hybrid graphical-simulation analysis of a health systems application*



**Barry Nelson**
Department of Industrial & Systems Engineering
*Ohio State University*
Columbus, Ohio 43210

**A. Ravindran**
School of Industrial Engineering
*University of Oklahoma*
Norman, Oklahoma 73019



BARRY NELSON is an assistant professor of industrial engineering at the Ohio State University in Columbus, Ohio. He received his MS (1981) and PhD (1983) degrees from Purdue University (West Lafayette, Ind.) in the School of Industrial Engineering. His research interests include statistical analysis of simulation experiments, particularly variance reduction techniques. Dr. Nelson teaches courses in computer simulation and stochastic processes.

A. RAVINDRAN is professor and director of the School of Industrial Engineering at the University of Oklahoma, Norman. He received his BS in electrical engineering from Birla Institute of Technology and Science, Pilani (India), and his MS and PhD degrees in industrial engineering and operations research from the University of California at Berkeley. His research interests include mathematical programming, multicriteria optimization, goal programming, metal cutting, health planning, energy models, and transportation analysis. His professional affiliations include IIE, ORSA, TIMS, ASEE, SME, Sigma Xi, and Alpha Pi Mu.

### ABSTRACT

*Computer simulation is often used to optimize practical network queuing models. In many situations the number of possible alternatives that could be simulated is large, and individual runs may be expensive. A simple approximation technique can be used to reduce the number of options actually simulated by providing a pictorial representation of the service process in a network of queues under simplifying assumptions. A simulation experiment can then be performed for detailed analysis of the selected subset of alternatives. This hybrid graphical-simulation approach was used in the design of a United States Air Force health improvement program.*

## INTRODUCTION

Computer simulation is often used to optimize practical network queuing models. Decision variables include staffing levels at service centers, customer routing patterns, queue disciplines, and load scheduling. In many situations the number of possible alternatives that could be simulated is large, and individual runs may be expensive. It is therefore desirable to simulate only a subset of likely candidates for the best alternative, whatever "best" is determined to be. Approximation, intuition, past experience with similar systems, and analytic results are common tools for presimulation analysis. For instance, factor screening[1] uses preliminary simulation runs to discover what decision variables have significant influence on system performance, and Jackson networks[5] are a broad class of analytically tractable queuing models that can be used to approximate the behavior of more general systems.

A simple approximation technique can be used to reduce the number of options actually simulated by providing a pictorial representation of the service process in a network of queues under simplifying assumptions. This technique is called GANS for Graphical Analysis of Network Structures. GANS is a simple, interactive computer procedure that provides information on the effects of different system designs based on few inputs; it does not necessarily yield accurate approximations of stochastic effects. The GANS approach is particularly useful for displaying system behavior over time, as opposed to summary statistics.

In the next section, background on graphical analysis as developed by Gordon Newell is given. The following section details the development of the GANS procedures, which are particularly efficient implementations of Newell's ideas. The combined use of graphical analysis and simulation in a real problem is the subject of AN APPLICATION: THE *HEART* PROGRAM; the last section presents some conclusions and possible extensions of the graphical approach.

## NEWELL GRAPHS

The GANS concept grew out of a body of work by Gordon Newell; an overview is given in Reference 9. Newell developed a flexible graphical technique for preliminary study of many practical queuing problems, particularly when stochastic properties of the queuing system are secondary to design effects. Let

$A_{ij}(t)$ = Cumulative number of customers of type $i$ to arrive at service point $j$ by time $t$

$D_{ij}(t)$ = Cumulative number of customers of type $i$ to depart from service point $j$ by time $t$.

*Research funded under contract F33615-80-C-0615 for the U.S. Air Force, School of Aerospace Medicine.

For any choices of $i$ and $j$ it is possible to construct step functions with unit steps at the arrival or departure times of each customer. The queue of customers at time $t$ is given by

$$Q_{ij} = A_{ij}(t) - D_{ij}(t)$$

If we also define

$A_{ij}^{-1}(k)$ = Time at which the $k$th customer of type $i$ arrives at service point $j$

$D_{ij}^{-1}(k)$ = Time at which the $k$th customer of type $i$ departs from point $j$

then the waiting time of the $k$th customer of type $i$ at point $j$ is

$$W_{ij}(k) = D_{ij}^{-1}(k) - A_{ij}^{-1}(k)$$

Figure 1 is an example of such a step function. Note that the total waiting time of all customers is the area between the two-step functions.

This is a general representation of almost all kinds of service systems. In fact, $A_{ij}(t)$ and $D_{ij}(t)$ may be interpreted as cumulative amounts of anything, provided it is conserved; this includes servers, who can be viewed as departing from a server pool to deliver service, and arriving back at the pool when service is completed. Analysis of queuing problems becomes the task of constructing $A_{ij}(t)$ and $D_{ij}(t)$ functions appropriate for the situation and observing conditions of congestion and idleness. The difficulty in constructing these graphs depends on the arrival and service processes and the system configuration. When simplifying assumptions, such as constant service time, can be made then iterative algorithms are possible. Several realizations can be constructed if the process is stochastic. A major feature of the graphical approach is the possibility of observing the evolution of the queuing process over time; its reaction to varying loads or transient conditions, for instance.

For example, in a single queue let $A_c(t)$ and $D_c(t)$ be the cumulative number of customers that have arrived and departed, respectively, by time $t$. Similarly, let $A_s(t)$ and $D_s(t)$ be the corresponding quantities for the servers. With $A_c(t)$ given and the service times specified, $D_s$, $A_s$ and $D_c$ can be derived. For instance, if $D_s^{-1}(k)$ is the time of the $k$th service start, and the $k$th service time is $s_k$, then $D_s^{-1}(k) + s_k$ is when the server rejoins the server queue. The $A_s$ function takes a unit jump at each such completion time. In the special case where all service times are constant and equal ($s_k = s$) the $A_s$ function is obtained as a simple translation of $D_s$ horizontally by $s$ and vertically by $n$, the number of servers; see Figure 2.

The next simplest case is where the $s_k$ are not equal, but $D_s^{-1}(k) + s_k$ is monotone increasing. In other words, there is some minimal variability, but servers cannot pass each other into service. Then the $A_s$ function is still a vertical translation of $D_s^{-1}(k) + s_k$. In general, however, the $D_s^{-1}(k) + s_k$ must be reordered to construct $A_s$. Algorithms for networks of queues are based on the fact that departures from one point in the network are arrivals to some other point; i.e., the departure function of one queue can be used as the arrival function of another queue.

Newell[8,9] shows how the graphical technique can be applied to GI/G/n queuing systems, while Newell[10] examines networks of finite capacity tandem queues. Hurdle[4] presents graphical analysis of a public transportation dispatching problem. Both authors show that under certain conditions these processes can be further approximated by diffusion equation models. The reader is encouraged to examine Gross and Harris[2] and Heyman and Sobel[3] for a different perspective on analyzing deterministic queuing systems.

In the papers by Newell and Hurdle the usefulness of graphical analysis is demonstrated. However, if one must construct the graphs by hand the approach becomes unattractive. GANS is a set of computer programs that produce graphs of queue evolution displaying information on customer and server queuing in a more compact and comprehensible form than the Newell graphs. Additional service functions not considered by Newell are also included in the GANS programs.

## GANS

GANS programs are written in BASIC-PLUS (with extend mode) and implemented on a Digital PDP-11 computer. The selection was made because of the desire for interactive program operation and the availability of the system at Purdue University. The programs are quite compact; so much so that versions have

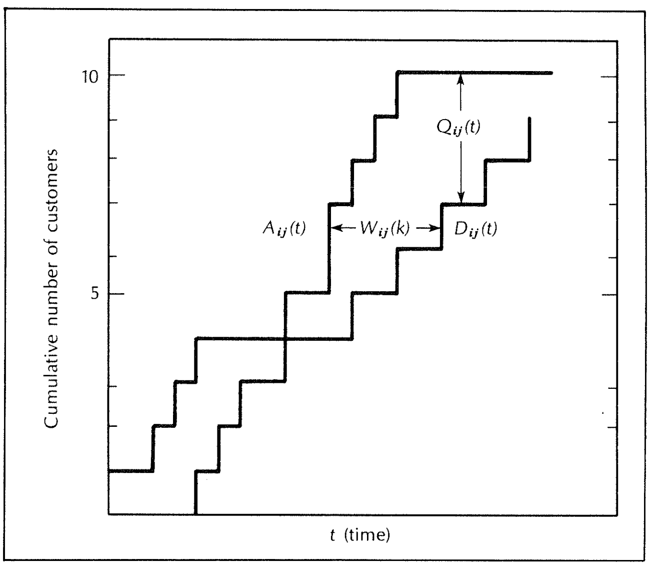



**Figure 1.** Customer arrival and departure times.

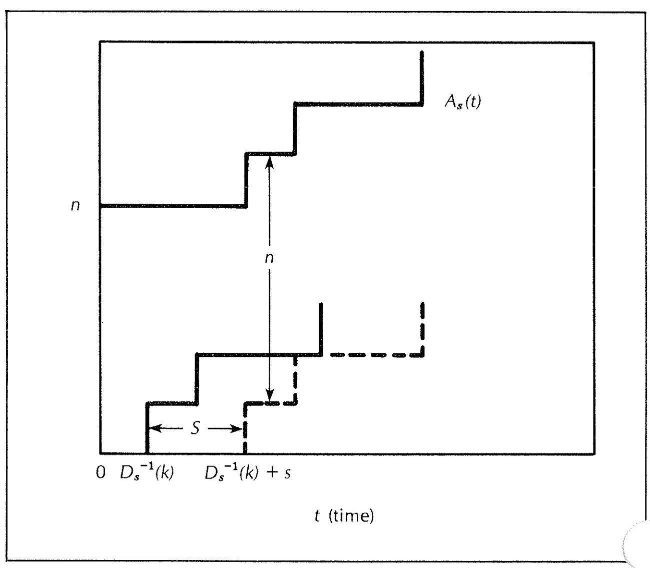



**Figure 2.** Derivation of server arrival times.

been implemented on systems as small as the Commodore "VIC 20" (with no additional memory). The programs are designed to be used at a terminal, so it seemed expedient to display graphical outputs at the user's terminal and not on a secondary plotting device. Thus, an important task was to redesign the Newell graphs so that they could easily be printed (or displayed) by a terminal and yet provide all necessary information at a glance.

First, it was decided to only permit integer units of time, meaning the smallest unit of time in use would be the standard. Although this may seem restrictive, GANS only provides a deterministic approximation so that precise service or arrival times would actually imply misleading accuracy. For the same reason, no summary statistics are computed.

Since the queues of customers and servers were of primary interest, the $A_c$ and $A_s$ graphs for each station were combined

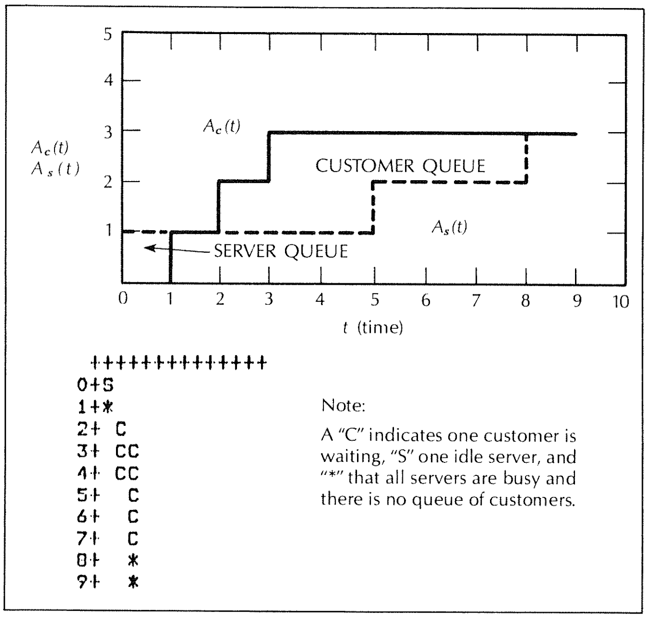



**Figure 3.** GANS output for a service station.

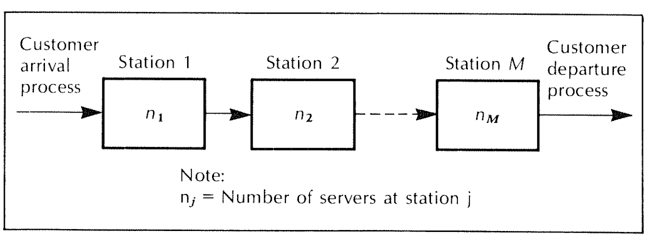



**Figure 4.** Example of a tandem queue.

in the following way: When $A_c(t)$ exceeds $A_s(t)$ then there is a queue of customers, and when the situation is reversed there are idle servers; $A_c(t) = A_s(t)$ means that all servers are busy and no customers are waiting. Figure 3 illustrates a GANS output for a station; thus, one graph displays all of the information for one service station.

The GANS programs were written to model two types of queuing networks (see Figures 4 and 5). Service times are assumed deterministic, although they may differ between stations, and from server to server within a station for the tandem queuing network (Figure 4). Queue capacity is unlimited and service is first-come-first-served, except at the feedback station where either new or returning customers may be given priority.

The GANS programs also include two service functions not considered by Newell, but that were particularly useful in the HEART Program (discussed in the next section). A *reset* time is a period following completion of a service when a server must recover or reset. An example would be paper work done after each service that does not prevent the customer from continuing. Additional delays in server availability occur because of *events*. A *regular event* happens every *m* customers, where *m* is specified for each station. These events can be used to model secondary responsibilities that occasionally, but predictably come up. On the other hand, an *unusual event* happens at most once, and can be used to model radical delays or breakdowns. The durations of resets and events are assumed constant and are specified for each station. An input file of customer arrival times to the first station is also required.

A detailed discussion of the GANS programs will not be given here,[6] but we will provide a brief summary of the logic. The tandem network program uses customer departures from one station as arrivals for just one other station. Thus, it can compute departure times for all customers arriving to a station, collect and reorder them, and then repeat the procedure for the next station. The program stores the ordered arrival times and times when servers will next be available and uses them to construct the graphical outputs. The feedback network program processes customers one-at-a-time through the network, necessitated by the addition of the feedback queue (Figure 5). Arrival times of returning customers must be compared to new arrivals to make sure that service is given in the proper order. The one-at-a-time approach requires the assumption of equal service times for all servers in a station to maintain the correct order of departures.

## AN APPLICATION: THE *HEART* PROGRAM

The Health Evaluation and Risk Tabulation (HEART) Program is a comprehensive preventative medicine program planned by the United States Air Force (USAF) to reduce the risk of cardiovascular disease among USAF personnel. As currently envisioned, each USAF base (118 total) would have a HEART program facility and staff to administer the program. One of the primary tasks of the HEART program staff is to screen USAF

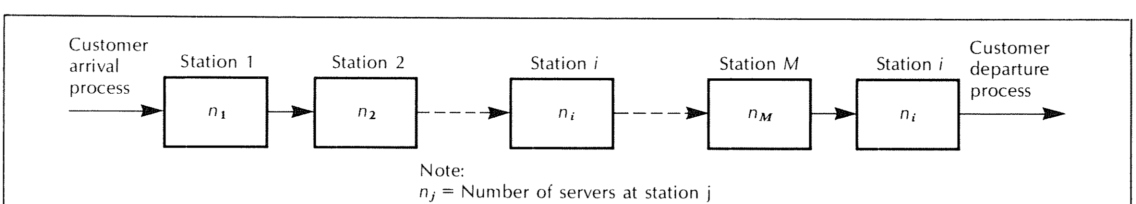



**Figure 5.** Example of a feedback queue.

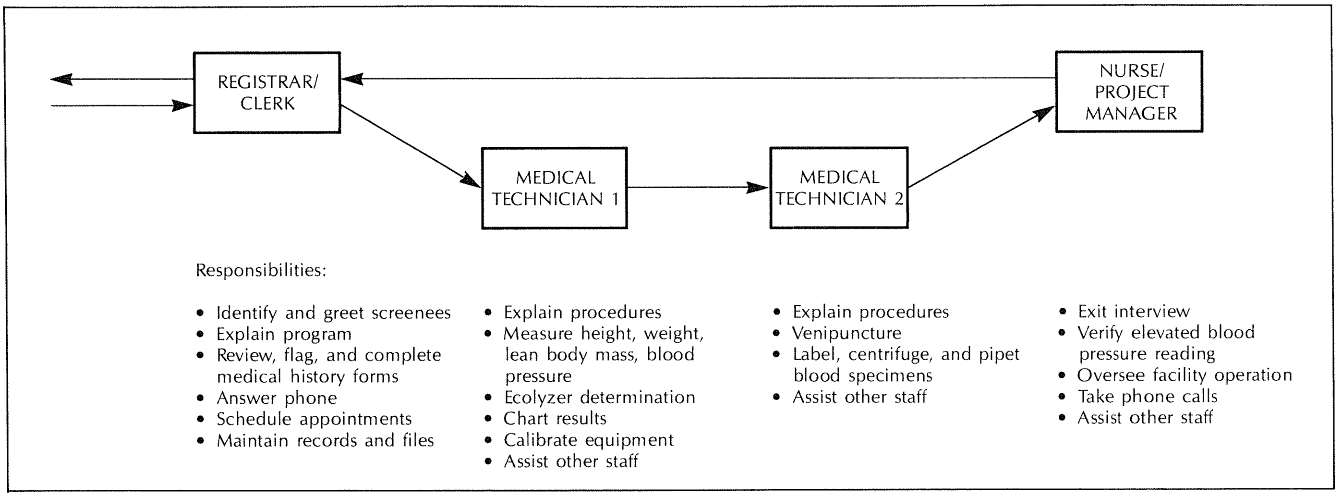



**Figure 6.** Schematic of cardiovascular risk screening program.

personnel to estimate their risk of cardiovascular disease. A simplified schematic of risk screening for a particular trial program is shown in Figure 6.

For additional details about the HEART Program, see Ravindran.[12]

The Air Force wanted a staffing plan for each base that would minimize the size of the staff while still meeting guidelines for task assignments, minimum performance standards, and certain operational procedures. A discrete-event simulation experiment, programmed in SLAM,[11] was eventually developed to analyze risk screening for different staff configurations under various screening loads. Transient results, such as congestion in the HEART facility throughout the day, were of particular interest to the USAF. The simulation experiment is discussed in Nelson[6] and Nelson and Ravindran.[7]

While data collection was planned and carried out, there was a need to provide the Air Force with some preliminary analysis and recommendations based on a small sample of observations of a trial program. The USAF had a preconceived idea of how staffing should be done, so it was necessary to give convincing arguments for radical departures from their plans. Also, since there were few restrictions on screening schemes and appointment scheduling algorithms, there were many potential options to simulate. For all of these reasons, GANS was developed.

As an evaluation tool, GANS contributed to the analysis of appointment scheduling algorithms and strategies for handling secondary screening responsibilities. For example, in the scenario shown in Figure 6 the second medical technician must centrifuge and pipet blood samples as well as take blood samples from USAF personnel. Alternatives include processing blood samples during idle time, when a certain number of vials are collected, or a combination of these two procedures; it is not possible to wait until the end of the screening day. Using records of arrival times from the trial program, and service times estimated from a few observations, GANS was used to evaluate the impact of various strategies on queue buildup. Regular events were used to model blood sample processing at regular intervals. It was discovered that not enough idle time existed to wait for idle periods, and that a threshold of 2 vials, but no more than 12 vials, should be collected before centrifuging and pipetting take place.

GANS was also useful for demonstrating that the two medical technicians (see Figure 6) should work in parallel rather than series; in other words, form one screening station with both technicians performing the same combined screening tasks. Early observations of risk screening revealed that the service time for the second technician was significantly shorter than the service time for the first one. Also, if one technician is interrupted (such as by a patient who faints when blood is drawn), normal patient flow is only slowed, not stopped, in the parallel configuration. Although the change seems obvious to those familiar with analysis of queuing networks, the GANS graphs provided a convincing way to demonstrate the logic of the parallel system to the Air Force.

For the demonstration, actual arrival times from some randomly selected days at the trial base were combined with rough estimates of service times. Unusual events were used to model screening interruptions and regular events for such things as pipetting blood samples. To illustrate, Figure 7 shows a GANS input and output segment for stations two and three in the series system, while Figure 8 gives the corresponding results for the parallel configuration. The reduced congestion (Station Two) and idleness (Station Three) are easily noted. Figures 9 and 10 show a similar comparison of the two systems when there is a major interruption. The interruption was modeled as an unusual event occurring after the fifth patient. It is clear that the congestion is minimal in the parallel system. The simulation analysis later validated these results.

## CONCLUSIONS AND RECOMMENDATIONS

There are many potential changes or additions to the GANS programs that would make it more versatile. In order to consider some stochastic elements in the service process, it would be relatively easy to permit input files of service times for the stations. It would also be possible to include some limited kinds of probabilistic branching. Current microcomputer graphics capabilities could be employed to draw sharper graphs and eliminate the need for integer time units; this would be at the expense of program size and speed, and would certainly make the programs less portable.

Preliminary graphical analysis can be done quickly, inexpensively, and with a minimum amount of information about the system under study. GANS can be used by researchers not

```
GANS PROGRAM INPUT -- FEEDBACK VERSION
NUMBER OF STATIONS (<=4)? 4


STATION DATA INPUT

FOR STATION  1
NUMBER OF SERVERS? 1
SERVICE TIME? 4
RESET TIME BETWEEN SERVICES? 1
ARE THERE ANY REGULAR EVENTS (YES OR NO)? NO
IS THERE AN UNUSUAL EVENT (YES OR NO)? NO

FOR STATION  2
NUMBER OF SERVERS? 1
SERVICE TIME? 10
RESET TIME BETWEEN SERVICES? 0
ARE THERE ANY REGULAR EVENTS (YES OR NO)? NO
IS THERE AN UNUSUAL EVENT (YES OR NO)? NO

FOR STATION  3
NUMBER OF SERVERS? 1
SERVICE TIME? 5
RESET TIME BETWEEN SERVICES? 0
ARE THERE ANY REGULAR EVENTS (YES OR NO)? YES
AFTER HOW MANY CUSTOMERS? 2
DURATION OF EVENT? 2
IS THERE AN UNUSUAL EVENT (YES OR NO)? NO

FOR STATION  4
NUMBER OF SERVERS? 1
SERVICE TIME? 6
RESET TIME BETWEEN SERVICES? 1
ARE THERE ANY REGULAR EVENTS (YES OR NO)? NO
IS THERE AN UNUSUAL EVENT (YES OR NO)? NO


FEEDBACK STATION INPUT

STATION NUMBER OF FEEDBACK STATION? 1
SERVICE TIME? 1
RESET TIME? 0
ANY REGULAR EVENTS? NO
IS THERE AN UNUSUAL EVENT? NO


ARRIVAL FILE NAME? DAY13.HAF



(Station Two)
 70+       C
 71+       C
 72+       C
 73+       *
 74+        C
 75+        C
 76+        C
 77+        C
 78+        C
 79+        C
 80+        C
 81+        CC
 82+        CC
 83+         C
 84+         C
 85+         C
 86+         C
 87+         C
 88+         C
 89+         C
 90+         CC
 91+         CC
 92+         CC
 93+          C
 94+          C
 95+          CC
 96+          CC
 97+          CC
 98+          CC

(Station Two, Continued)
 99+           CC
100+           CC
101+           CCC
102+           CCC
103+            CC
104+            CC
105+            CC
106+            CCC
107+            CCC
108+            CCC
109+            CCC
110+            CCC
111+            CCC
112+            CCC
113+             CC
114+             CC
115+             CC
116+             CC
117+             CC
118+             CC
119+             CC
120+             CC

(Station Three)
 70+       S
 71+       S
 72+       S
 73+       *
 74+       *
 75+       *
 76+       *
 77+       *
 78+       *
 79+       *.
 80+        S
 81+        S
 82+        S
 83+        *
 84+        *
 85+        *
 86+        *
 87+        *
 88+         S
 89+         S
 90+         S
 91+         S
 92+         S
 93+         *
 94+         *
 95+         *
 96+         *
 97+         *
 98+         *
 99+         *
100+          S
101+          S
102+          S
103+          *
104+          *
105+          *
106+          *
107+          *
108+           S
109+           S
110+           S
111+           S
112+           S
113+           *
114+           *
115+           *
116+           *
117+           *
118+           *
119+           *
120+            S
```

**Figure 7.** GANS input-output segment with Stations 2 and 3 in series.

```
GANS PROGRAM INPUT -- FEEDBACK VERSION


NUMBER OF STATIONS (<=4)? 3


STATION DATA INPUT

FOR STATION  1
NUMBER OF SERVERS? 1
SERVICE TIME? 4
RESET TIME BETWEEN SERVICES? 1
ARE THERE ANY REGULAR EVENTS (YES OR NO)? NO
IS THERE AN UNUSUAL EVENT (YES OR NO)? NO

FOR STATION  2
NUMBER OF SERVERS? 2
SERVICE TIME? 15
RESET TIME BETWEEN SERVICES? 0
ARE THERE ANY REGULAR EVENTS (YES OR NO)? YES
AFTER HOW MANY CUSTOMERS? 2
DURATION OF EVENT? 2
IS THERE AN UNUSUAL EVENT (YES OR NO)? NO

FOR STATION  3
NUMBER OF SERVERS? 1
SERVICE TIME? 6
RESET TIME BETWEEN SERVICES? 1
ARE THERE ANY REGULAR EVENTS (YES OR NO)? NO
IS THERE AN UNUSUAL EVENT (YES OR NO)? NO


FEEDBACK STATION INPUT

STATION NUMBER OF FEEDBACK STATION? 1
SERVICE TIME? 1
RESET TIME? 0
ANY REGULAR EVENTS? NO
IS THERE AN UNUSUAL EVENT? NO


ARRIVAL FILE NAME? DAY13.HAF

(Station Two)
 70+       *
 71+       *
 72+       *
 73+       *
 74+        C
 75+        C
 76+        *
 77+        *
 78+        *
 79+        *
 80+        *
 81+         C
 82+         C
 83+         *
 84+         *
 85+         *
 86+         *
 87+         *
 88+         *
 89+         *
 90+          C
 91+          C
 92+          C
 93+          *
 94+          *
 95+          *
 96+           C
 97+           C
 98+           *
 99+           *
100+           *
101+           *
102+            C
103+            C
104+            C
105+            C
106+            C
107+            CC
108+            CC
109+            CC
110+             C
111+             C
112+             C
113+             *
114+             *
115+             *
116+             *
117+             *
118+             *
119+             *
120+             *
```

**Figure 8.** GANS input-output segment with Stations 2 and 3 in parallel.

```
GANS PROGRAM INPUT -- FEEDBACK VERSION
NUMBER OF STATIONS (<=4)? 4


STATION DATA INPUT

FOR STATION  1
NUMBER OF SERVERS? 1
SERVICE TIME? 4
RESET TIME BETWEEN SERVICES? 1
ARE THERE ANY REGULAR EVENTS (YES OR NO)?
IS THERE AN UNUSUAL EVENT (YES OR NO)?

FOR STATION  2
NUMBER OF SERVERS? 1
SERVICE TIME? 7
RESET TIME BETWEEN SERVICES? 0
ARE THERE ANY REGULAR EVENTS (YES OR NO)?
IS THERE AN UNUSUAL EVENT (YES OR NO)?

FOR STATION  3
NUMBER OF SERVERS? 1
SERVICE TIME? 4
RESET TIME BETWEEN SERVICES? 0
ARE THERE ANY REGULAR EVENTS (YES OR NO)? YES
AFTER HOW MANY CUSTOMERS? 2
DURATION OF EVENT? 2
IS THERE AN UNUSUAL EVENT (YES OR NO)? YES
AFTER HOW MANY CUSTOMERS? 5
DURATION OF EVENT? 12

FOR STATION  4
NUMBER OF SERVERS? 1
SERVICE TIME? 6
RESET TIME BETWEEN SERVICES? 1
ARE THERE ANY REGULAR EVENTS (YES OR NO)? NO
IS THERE AN UNUSUAL EVENT (YES OR NO)? NO


FEEDBACK STATION INPUT

STATION NUMBER OF FEEDBACK STATION? 1
SERVICE TIME? 1
RESET TIME? 0
ANY REGULAR EVENTS?
IS THERE AN UNUSUAL EVENT?


ARRIVAL FILE NAME? DAY13.HAF

(Station Three)
 58+       S
 59+       S
 60+       *
 61+       *
 62+       *
 63+       *
 64+       *
 65+       *
 66+       *
 67+        C
 68+        C
 69+        C
 70+        C
 71+        C
 72+        C
 73+        C
 74+        C
 75+        CC
 76+         C
 77+         C
 78+         C
 79+         C
 80+         C
 81+         C
 82+          C
 83+          C
 84+          C
 85+          C
 86+          *
 87+          *
 88+          *
 89+           C
 90+           C
 91+           C
 92+           *
 93+           *
 94+           *
 95+           *
 96+            S
 97+            S
 98+            *
 99+            *
100+            *
```

**Figure 9.** GANS input-output segment for a series system with a major interruption.

familiar or experienced with simulation modeling, since the programs prompt the user for all inputs. A major drawback, of course, is the assumption of deterministic service times, which is why the graphical technique is best combined with simulation to validate results. If the effects of variability are the primary concern, then analysis by GANS might be deceiving.

Perhaps the greatest contribution that graphical analysis can make is as a tool for exposition. It takes very little time to learn how to read the graphs and they contain significant information about how a queuing system behaves over time. An unexplored, but potentially valuable application for Newell type graphs is as a means for displaying simulation results, particularly when analysis of congestion is of interest. Summary statistics and histograms tend to cover up time dependent phenomena, while GANS outputs illustrate how alternative

```
GANS PROGRAM INPUT -- FEEDBACK VERSION


NUMBER OF STATIONS (<=4)? 3


STATION DATA INPUT

FOR STATION  1
NUMBER OF SERVERS? 1
SERVICE TIME? 4
RESET TIME BETWEEN SERVICES?     1
ARE THERE ANY REGULAR EVENTS (YES OR NO)? NO
IS THERE AN UNUSUAL EVENT (YES OR NO)? NO

FOR STATION  2
NUMBER OF SERVERS? 2
SERVICE TIME? 11
RESET TIME BETWEEN SERVICES? 0
ARE THERE ANY REGULAR EVENTS (YES OR NO)? YES
AFTER HOW MANY CUSTOMERS? 2
DURATION OF EVENT? 2
IS THERE AN UNUSUAL EVENT (YES OR NO)? YES
AFTER HOW MANY CUSTOMERS? 5
DURATION OF EVENT? 12

FOR STATION  3
NUMBER OF SERVERS? 1
SERVICE TIME? 6
RESET TIME BETWEEN SERVICES? 1
ARE THERE ANY REGULAR EVENTS (YES OR NO)?
IS THERE AN UNUSUAL EVENT (YES OR NO)?


FEEDBACK STATION INPUT

STATION NUMBER OF FEEDBACK STATION? 1
SERVICE TIME? 1
RESET TIME? 0
ANY REGULAR EVENTS?
IS THERE AN UNUSUAL EVENT?


ARRIVAL FILE NAME? DAY13.HAF
```

```
(Station Two)

 58+     S
 59+     *
 60+     *
 61+     *
 62+     *
 63+     *
 64+     *
 65+     *
 66+     *
 67+     *
 68+      C
 69+      C
 70+      C
 71+      C
 72+      *
 73+      *
 74+      *
 75+       C
 76+       *
 77+       *
 78+       *
 79+       *
 80+       *
 81+       *
 82+        C
 83+        *
 84+        *
 85+        *
 86+        *
 87+        *
 88+        *
 89+         S
 90+         *
 91+         *
 92+         *
 93+         *
 94+          S
 95+          S
 96+          *
 97+          *
 98+          *
 99+          *
100+          *
```

**Figure 10.** GANS input-output segment for a parallel system with a major interruption.

systems perform during peak periods or under unexpected workloads. By combining graphical analysis and simulation, the full advantages of both may be realized.

## REFERENCES


1 BILES, W. E. and SWAIN, J. J.
*Optimization and Industrial Experimentation*. John Wiley & Sons, New York, N.Y. (1980).

2 GROSS, D. and HARRIS, C. M.
*Fundamentals of Queuing Theory*. John Wiley & Sons, New York, N.Y. (1974).

3 HEYMAN, D. P. and SOBEL, M. J.
*Stochastic Models in Operations Research, Volume I*. McGraw-Hill Book Co., New York, N.Y. (1982).

4 HURDLE, V. F.
"Minimum Cost Schedules for a Public Transportation Route, I Theory, II Examples." *Transportation Science* 7 (1973), 109-157.

5 JACKSON, J. R.
"Networks of Waiting Lines." *Operations Research* 5 (1957), 518-521.

6 NELSON, B. L.
"Combined Graphical-Simulation Analysis of a USAF Staffing Problem." Unpublished Master's thesis. School of Industrial Engineering, Purdue University, West Lafayette, Ind. (1981).

7 NELSON, B. L. and RAVINDRAN, A.
"A Hybrid Graphical-Simulation Analysis of a Staffing Problem in the U.S. Air Force." *Research Memorandum No. 82-2*, School of Industrial Engineering, Purdue University, West Lafayette, Ind. (1982).

8 NEWELL, G. F.
"Approximate Behavior of n-Server Systems with Large n." *Lecture Notes in Economics and Mathematical Systems* 87 (1973).

9 NEWELL, G. F.
"Graphical Representation of Queue Evolution for Multiple Server Systems." *Lecture Notes in Economics and Mathematical Systems* 98 (1974).

10 NEWELL, G. F.
"Approximate Behavior of Tandem Queues." *Lecture Notes in Economics and Mathematical Systems* 171 (1979).

11 PRITSKER, A. A. B. and PEGDEN, C. D.
*Introduction to Simulation and SLAM*. John Wiley and Sons, New York, N.Y. (1979).

12 RAVINDRAN, A. *et al.*
"Using Simulation in Cardiovascular Disease Control." *Simulation*, 35:1 (1980), 13-20.